Consider the Shape class below that is similar to the one developed in lecture:

```
+-----------------------------------------------------+
|                          Shape                      |
+-----------------------------------------------------+
| -color: Color                                       |
| -centerLocation: Point2D                            |
+-----------------------------------------------------+
| +Shape(color: Color, centerX: double, centerY: double) |
| +getArea(): double                                  |
| +toString(): String                                 |
+-----------------------------------------------------+
```

Suppose that the getArea() method just returns zero and the toString() method returns something like this:

```
Color: 0x00ffffff
Location: Point2D [x = 0.0, y = 0.0]
```

Provide a complete implementation (excluding `package` or `import` statements) for a subclass called RightTriangle. The class should model a right triangle that oriented in any arbitrary direction (see examples drawn on whiteboard).

Suppose First is an interface with two methods: iMethodA() and iMethodB(). Suppose further that Parent is a class with the following methods: a constructor that accepts a String as an argument, an overridden equals() method, and pMethodA().

(a) Show the class declaration (just the first line) of a Child that inherits from Parent and implements the First interface.

(b) List the method signatures for all of the methods that **must** be implemented by the Child class.

(c) List all of the reference types that can refer to a Child object. I.e., what can replace XXX in the following line of code?

```
XXX obj = new Child();
```

Suppose First is an interface with two methods: iMethodA() and iMethodB(). Suppose further that Parent is a class with the following methods: a constructor that accepts a String as an argument, an overridden equals() method, and pMethodA().

(a) Show the class declaration (just the first line) of a Child that inherits from Parent and implements the First interface.

(b) List the method signatures for all of the methods that **must** be implemented by the Child class.

(c) List all of the reference types that can refer to a Child object. I.e., what can replace XXX in the following line of code?

```
XXX obj = new Child();
```

(a) Sketch what will be displayed when a class that extends Application contains the following method implementation:

```
@Override
public void start(Stage stage) {
    Pane root = new VBox();
    root.getChildren().addAll(
        new Label("This seems like"),
        new Button("a"),
        new Button("useless"),
        new Label("graphical"),
        new Button("user"),
        new Label("interface."));
    stage.setTitle("Quiz 4");
    stage.setScene(new Scene(root, 400, 400);
    stage.show();
}
```

(b) List two replacements for VBox in the first line of the start() method above and indicate how the UI would change for each replacement.

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    try {
        int i = in.nextInt();
        double x = -1.0;
        System.out.println(i*x);
        x = in.nextInt();
        System.out.println(i*x);
    } catch(RuntimeException e) {
        System.out.println("ouch");
    } finally {
        System.out.println("done");
    }
    System.out.println("psych");
}
```

Consider the code above and indicate, for each part, what will be displayed if the user were to enter the following input.

(a)

```
3 5
```

(b)

```
thirty three
```

(c)

```
8 30.0
```

Complete the program below that writes the values entered by the user to two files: quiz7.bin and quiz7.txt. The first file should be written using a DataOutputStream and the second file should be written using a PrintWriter.

```
public static void main(String[] ignored) {
    Scanner in = new Scanner(System.in);
    System.out.println("Please␣enter␣the␣temperature␣to␣three␣significant␣figures␣(e.g.,␣72.8)");
    double first = in.nextDouble();
    System.out.println("Please␣enter␣your␣age␣in␣years␣(e.g.,␣18");
    int second = in.nextInt();
    // ...
```

(a) What is a functional interface and why is it useful?

(b) Supposed numbers is a List<String> use functional programming techniques to determine the number of unique strings in the list.

(a) Define **affordance** and give an example of how it can be used in UI design.

(b) Given a list of integers, return a list where each integer is added to 1 and the result is multiplied by 10.

```
// math1([1, 2, 3]) -> [20, 30, 40]
// math1([6, 8, 6, 8, 1]) -> [70, 90, 70, 90, 20]
// math1([10]) -> [110]
public static List<Integer> math1(List<Integer> nums) {
```